An adaptive model for 2D and 3D dense non rigid motion computation

Serge Benayoun, Nicholas Ayache, Isaac Cohen

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An adaptive model for $2D$ and $3D$ dense non rigid motion computation

Serge Benayoun & Nicholas Ayache & Isaac Cohen

N° 2297

Mai 1994

PROGRAMME 4

Robotique,

image

et vision

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

An adaptive model for $2D$ and $3D$ dense non rigid motion computation

Serge Benayoun & Nicholas Ayache & Isaac Cohen *

Programme 4 — Robotique, image et vision
Projet Epidaure

Rapport de recherche n° 2297 — Mai 1994 — 28 pages

Abstract: We describe a new method for computing a dense displacement field from a time-sequence of $2D$ or $3D$ images. It consists in minimizing an energy defined on the space of correspondence functions. This energy is divided into two terms, one term which matches contour points and particularly high curvature points, and one regularization term which constrains the continuity of the field. We introduce an adaptive mesh the resolution of which depends on the presence of edges and/or points of high curvature and we show how to use it to reduce the computational time of the method. We present experimental results on medical images which prove the validity of the approach and the accuracy of the computed displacement fields.

Key-words: Motion field, Differential singularities, Finite element method, Adaptive Meshes, Medical Imaging, $3D$.

(Résumé : tsvp)

*Projet Masda, INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

Un modèle adaptatif pour le calcul du mouvement dense et non rigide

Résumé : Nous décrivons une méthode originale de calcul de champ dense de vecteurs de déplacement sur des séquences temporelles d'images $2D$ ou $3D$. Elle consiste à minimiser une énergie définie sur l'espace des fonctions de correspondance. Cette énergie comporte deux termes, un terme qui vise à apparier les points de fort gradient dans l'image, et particulièrement les points de forte courbure, et un terme de régularisation qui contrôle la continuité du champ des vecteurs. Nous introduisons ensuite la technique des maillages adaptatifs pour laquelle la densité des noeuds dépendra de la présence de points de contour et/ou de points de forte courbure dans l'image. Nous montrons comment utiliser cette approche pour réduire le temps de calcul du champ des déplacements. Nous présentons enfin des résultats expérimentaux sur des images médicales qui illustrent la validité de notre approche et la précision du champ de vecteurs ainsi calculé.

Mots-clé : Champ dense de vecteurs de déplacement, Singularités différentielles, Méthode des éléments finis,Maillages adaptatifs, Imagerie Médicale, $3D$.

1 Introduction

The problem of the computation of non rigid motion field from a time sequence of images is one of the most important challenge in computer vision. There are many applications like model-based image compression [TH94] or medical diagnosis [Aya93]. In medical imaging processing such sequences are important since they provide both anatomical and physiological information. A particularly important application concerns the analysis of the cardiac deformations [AD92, SD92, HG93b, CHA94, KG94]. Three-dimensional ($3D$) medical imagery allows to recover the complete dynamic of the left ventricle wall. Furthermore, the time resolution permits a precise study of the heart beats, not limited to diastole and systole. So it is now conceivable to compute accurately the left ventricle wall motion parameters. Such a result can help the cardiologist, for example to evaluate the effect of a treatment during several months.

In this paper we present a new method for computing non rigid motion fields in a temporal sequence of $2D$ or $3D$ medical images. Our method is fully automatic, we do not use markers to track the motion. Furthermore the structures appearing in these images are non rigid and tend to present specific salient geometric features. In the following we present a method allowing to incorporate such an information.

We pursue the original idea (introduced in a previous paper [CAS92]) of using geometric singularities to anchor the motion field on a few but reliable characteristic points, and to propagate this sparse motion field to contours first, and then to every point. The difference from [CAS92] is that we can deal with an arbitrary number of possibly open contours in the images, and we end up with a dense motion field computed everywhere, not limited to the points of a single closed contour. The idea of using salient features to guide the components of the non rigid motion field was also mentioned in different studies reported by [DWS83, BK89, AD92].

The way we computationally implement our method is by an original energy minimization formulation, where three terms represent exactly the constraints imposed respectively to singular points, contour points, and regular points, with adequate weighting. This minimization is done with a mathematically rigorous finite element method (section 2). This technique allows us to intro-

duce an adaptive mesh, the resolution of which depends on the presence of high gradient norm points and/or high curvature points. Such an idea follows prior studies reported by [TV91, HG93a]. This allows us to reduce the computational time without decreasing the accuracy of the displacement fields near salient features (section 3). We show finally experimental results on synthetic data, and real medical images of articulated and deformable structures which prove the validity of our approach (section 4).

2 Computing a Dense Motion Field

2.1 Energy Formulation

Medical imagery produces images which describe the temporal evolution of anatomic structures. Magnetic Resonance Imagery (MRI), Ultrasound Imagery, Nuclear Medicine Imagery and even Computed Tomography Imagery produce time sequence images, for example of the cardiac cycle [Aya93]. Processing such sequences allows the computation of the motion of the cavities. In this section we present a method which computes the displacement field over the whole image, without any prior segmentation.

The determination of the non rigid motion field is based on matching points between two consecutive frames, which have similar differential characteristics. We use two types of attributes :

- High gradient norm points : contour points correspond to local extrema of the gradient norm of the grey level function, in the direction of the intensity gradient.
- High curvature points : these points are used as landmarks to guide the matching between similar points.

External Energy These two attributes allow to guide the matching of high gradient norm points for which the geometric characterization given by high curvature points is relevant. These correspondences are achieved by minimizing an *external energy* E_e split into two terms, a *curvature dependent energy* E_c:

$$E_c(f) = \int_{\Omega_t} (C_t - C_{t+1}(f(x,y)))^2 R_c(x,y)dxdy = \int_{\Omega_t} e_c^2(x,y)R_c(x,y)dxdy \quad (1)$$

where Ω_t is the domain of the first image, C_t is the Euclidean curvature at time t, R_c is a weighting function and f is the correspondance function, and a *gradient dependent energy* E_g:

$$E_g(f) = \int_{\Omega_t} (N_t - N_{t+1}(f(x,y)))^2 R_g(x,y)dxdy = \int_{\Omega_t} e_g^2(x,y)R_g(x,y)dxdy \quad (2)$$

where N_t is the gradient norm at time t and R_g is a weighting function.

Internal Energy The minimization of the previous energy is an ill-posed problem. There is indeed no single solution. We add also an *internal energy* E_i which constrains the regularity of the solution :

$$E_i(f) = \int_{\Omega_t} \|\nabla f(x,y)\|^2 dxdy \quad (3)$$

It measures the variation of the displacement field over the whole image and constrains the solution to be C^0 continuous.

Weighting Functions The internal and external energies are balanced through two *weighting functions* R_g and R_c. R_g (resp. R_c) is an increasing normalization function of the gradient norm (resp. curvature). These weighting functions enforce locally the external energies if relevant points are not well matched. Inversely, the weighting functions inhibit the external energies when a point has no significant features.We have chosen the polynomial threshold function acting like a high pass filter:

$$h(x) = -2\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right)^3 + 3\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right)^2, \quad (4)$$

where x_{min} and x_{max} are the bounding values.

We combine the weighting functions and we obtain a single expression for the external energy :

$$E_e(f) = \int_{\Omega_t} (1 - R_c) R_g (N_t - N_{t+1}(f))^2 + R_c (C_t - C_{t+1}(f))^2 dxdy \qquad (5)$$

This combination is justified for high curvature points. Indeed high curvature points have also high gradient norm. We prefer in this case to privilege the attraction by an other high curvature point (see figure 1).

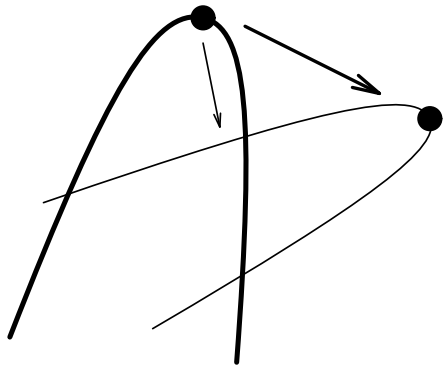

Figure 1: *Preferential attraction for high curvature points.*

2.2 Numerical solution

2.2.1 Partial Differential Equation

A necessary condition of optimality of the correspondence function $f = (u, v)$ is the Euler-Lagrange equation $\nabla E(f) = 0$ which is equivalent to :

$$\begin{cases} -\Delta f = (1 - R_c) R_g \nabla N_{t+1}(f) e_g + R_c \nabla C_{t+1}(f) e_c \\ f_{/\partial\Omega} = f_0 \text{ (boundary conditions)} \end{cases} \qquad (6)$$

We obtain also a system of two partial differential equations, one for each component of the displacement vectors field :

$$\begin{cases} -\Delta u = (1-R_c)R_g\nabla\dfrac{\partial N_{t+1}}{\partial x}(f)e_g + R_c\nabla\dfrac{\partial C_{t+1}}{\partial x}(f)e_c \\ -\Delta v = (1-R_c)R_g\nabla\dfrac{\partial N_{t+1}}{\partial y}(f)e_g + R_c\nabla\dfrac{\partial C_{t+1}}{\partial y}(f)e_c \end{cases} \tag{7}$$

The solution of each equation is performed through a variational method. Considering the first equation consists to solve :

$$\begin{aligned} &\text{find } u \in H^1(\Omega_t) = \{w \in L^2(\Omega_t)/\partial w \in L^2(\Omega_t)\} \text{ such that} \\ &\qquad a(u,w) = L_{(u,v)}(w)\ ,\ \forall w \in H^1(\Omega_t), \end{aligned} \tag{8}$$

where

$$\begin{aligned} a(u,w) &= \int_{\Omega_t} \nabla u \nabla w dxdy \\ L_{(u,v)}(w) &= \int_{\partial\Omega_t} (\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}) w d\sigma + \\ &\qquad \int_{\Omega_t} ((1-R_c)R_g\frac{\partial N_{t+1}}{\partial x}e_g + R_c\frac{\partial C_{t+1}}{\partial x}e_c) w dxdy \end{aligned}$$

A solution of this variational problem is difficult to achieve since the linear mapping L depends on the given data (the differential characteristics of the images) and also on the solution. Instead we consider the associated evolution problem allowing a linearization of L and characterizing the solution of equation (8) as a stationary solution of the evolution problem. The associated evolution equation is given by :

$$\begin{cases} \left(\dfrac{\partial u}{\partial t}, w\right) + a(u,w) = L_{(u,v)}(w)\ ,\ \forall w \in H^1(\Omega_t) \\ \text{with } (u_0, v_0) \text{ initial estimation} \end{cases} \tag{9}$$

The introduction of the evolution equation needs the definition of an initial displacement field f_0. The initial estimation is computed in two stages. The first stage consists in searching local correspondences between similar points. We match first high curvature points. If curvature is small, we match high gradient norm points. The second stage takes into account the sparse results

of the first procedure. We have indeed no initial displacement for points of low gradient norm. At these points we approximate the non rigid motion field by assuming that the motion is locally affine. We obtain the parameters of the affine transformation by least squares minimization. Thus we have an initial displacement vectors field which is dense and regular over the whole image.

2.2.2 Finite Element Method

The solution of the partial differential equation (9) is achieved by a finite element method to convert its continuous representation into a numerical representation. We use more precisely a finite difference scheme for the time discretization and a finite element scheme for the space discretization. For this purpose we consider a tessellation of the image domain Ω_t into a finite number of finite elements K. Each finite element is characterized by three features :

- *Shape:* triangles or rectangles.
- *Family of Polynomial Functions:* P^1 for triangular shape, Q^1 for rectangular shape.
- *Set of degrees of freedom:* The value of the function at each vertex.

We need only a $C^0(\Omega_t)$ finite element since we have [Cia87] :

$$v \in C^m(\overline{\Omega}), v/_K \in C^{m+1}(K) \Rightarrow v \in H^{m+1}(\Omega)$$

A solution in the set of polynomial functions is obtained through a decomposition of the solution in the associated functions basis $\Phi_i, i = 1..n$ (where n is the number of vertices), leading to the iterative solution of a set of linear equations:

$$(I + \Delta t A)u^{t+1} = \Delta t \; L_{(u^t, v^t)} + u^t \tag{10}$$

where Δt is the time step, I is the identity matrix, $A = (a(\Phi_i, \Phi_j))_{i,j=1..n}$ is the stiffness matrix, L the external forces vector and $f^t = (u^t, v^t)$ the displacement field at time t.

The computational time for the convergence is mainly dependent on the size n of the discretization of the domain Ω_t. It is the reason why we introduce now the notion of adaptive mesh.

3 Adaptive Meshes

3.1 Motivation

The size of the linear system (10) corresponds to the number of nodes in the discretization of the image domain Ω_t. This computation can also be speeded-up by reducing the number of nodes.

The accuracy of the computed displacement field depends on the number of points tessellating the domain Ω_t. The more accurate solution is obtained when we consider a node for each pixel in Ω_t leading to the iterative solution of a large linear system. In order to reduce the size of this system, we have first tried a *multiresolution scheme* [WTK87]. It consists in computing first the solution obtained at a low resolution, for example one node each four pixels. We use next this solution as an initial estimate of the solution for higher resolution meshes (see figure 2).

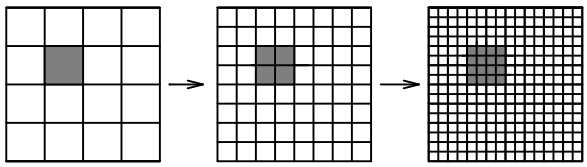

Figure 2: *Multiresolution Scheme.*

The idea of this scheme is to converge roughly with the low resolutions which are not computationally expensive, and next to refine the results with a few iterations at the higher resolution. Nevertheless we need to return to the higher resolution, i.e. the pixel resolution, at the end of the convergence.This drawback is eliminated by using adaptive mesh. Indeed, this scheme consists in considering a mesh where the density of nodes is high near the relevant regions of the image, and on the opposite low for the background (see an example on figure 3).

The adaptive meshes are a classical tool in Fluid Dynamics [PD83, And86, DG93]. They have been already used in computer vision, in particular for image reconstruction [TV91], or surface tracking [HG93a]. The major advantage of

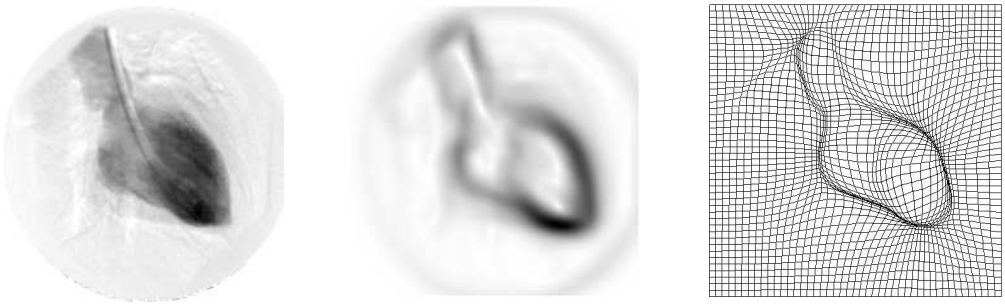

Figure 3: *Adaptive deformation of a mesh. On left, raw image. On middle, gradient norm of the image function from which the mesh is deformed.On right, adaptive mesh. We started with a regular mesh having one node each four pixels. The density of nodes is increased near contour points.*

this method is to decrease the numerical complexity without reducing the result accuracy.

There exist different methods to construct adaptive meshes. We will make the distinction between *unstructured mesh* and *structured mesh*. For example, we can initialize the mesh at the pixel resolution and eliminate progressively nodes which are not relevant. This is particularly useful for data compression problems. Nevertheless, this approach implies a computation of the adjacency graph of the nodes. It is indeed necessary for the finite element method. The type of connectivity between nodes is not preserved and the implementation of the finite element method is not easy. For this reason we have preferred to use a structured mesh where connectivity between nodes is preserved.

3.2 Definition

A *structured mesh* is a set of *nodes* connected by springs. Each spring modelizes the connection between two neighbor nodes by the well known relation :

$$\vec{S_{ij}} = k_{ij} d_{ij} \vec{X_{ij}} \tag{11}$$

The strength $\vec{S_{ij}}$ exerted by the node j to the node i is directed by the unit vector $\vec{X_{ij}}$ which links the two nodes. The intensity of the strength is proportional to the elongation of the spring d_{ij}, and also to the stiffness of the spring k_{ij}. The stiffness measures the intensity of the link.

A mesh is *adaptive* when the density of the nodes depends locally on the spatial position of the node. In our case the nodes evolve in the domain of an image. Our aim is to increase the density of nodes near the high gradient norm points of the image.

A first technique consists in the adaptation of the stiffness of the spring to the gradient norm of the two corresponding nodes [TV91]:

$$k_{ij} = \frac{\aleph(N_t(i)) + \aleph(N_t(j))}{2} \tag{12}$$

where $\aleph$ is a normalization function.

A second technique consists in the application of an external force. This force pulls the nodes toward the high gradient norm points. It can be derived from a distance map issued from a contour points binary image or from a potential field computed by differentiation of the gradient norm over the whole image. Using distance map needs yet a contour detection stage and next a nearest neighbor searching stage.

These two methods are very similar and we have implemented them in a same process where it is possible to combine them adequately.

3.3 Implementation

The adaptation of the mesh comes from the resolution of the following dynamic equation :

$$m_i \frac{\partial^2 X_i}{\partial t^2} + \gamma_i \frac{\partial X_i}{\partial t} = \vec{F_i} \ , \ i = 1..n \tag{13}$$

where n is the nodes number, X_i is the position of the node i, m_i its mass, γ_i its damping coefficient, and $\vec{F_i}$ the sum of the applied forces. We have chosen a unit mass for all the nodes except for boundary nodes which have infinite mass.

$\vec{F_i}$ is composed of two terms. First, a spring term $\vec{S_i}$ which modelizes the connection of the node with the set of its neighbors V_i (see section 3.2) :

$$\vec{S_i} = \sum_{j \in V_i} \vec{S_{ij}} \tag{14}$$

We consider here the 8-neighborhood to preserve the regularity of the mesh. Secondly, we have a image term $\vec{G_i}$ which attracts the node to the high gradient norm points and/or high curvature points :

$$\vec{G_i} = \nabla N_t(i) + \nabla C_t(i) \tag{15}$$

We have used a finite difference scheme to solve this system. The initial mesh is a regular mesh. In our experiments, we have chosen to consider a node each four or five pixels. The mesh deforms through the application of the forces and converges to an equilibrium.

3.4 Application to Motion problem

The finite element method yields the discretization of the domain of the correspondance functions in the considered space, here $H^1(\Omega_t)$. This discretization is given by the set of nodes $(n_i)_{i=1,n}$. We have chosen to tessellate the domain Ω_t into triangles. In order to approximate $H^1(\Omega_t)$, we need only a finite element of class $C^0(\Omega_t)$. Then, each basis function Φ_i is associated to a node n_i by the relation :

$$\Phi_i(n_j) = \delta_{ij} \tag{16}$$

where δ_{ij} is the Kronecker symbol.

Φ_i is also defined by parts on the six adjacent triangles $(T_k)_{k=1,6}$ to the node i (see figure 4). The elements of the stiffness matrix A are then obtained with the following computation :

$$\begin{aligned} a(i,j) &= \int_{\Omega_t} \nabla\Phi_i . \nabla\Phi_j \partial x \partial y \\ &= \sum_{k=1}^{6} \int_{T_k} \nabla\Phi_i^{(k)} . \nabla\Phi_j^{(k)} \partial x \partial y \end{aligned}$$

where $\Phi_i^{(k)}$ is the restriction of Φ_i to the triangle T_k.

This computation is made after a transformation of the deformed triangles $(T_k)_{k=1,6}$ onto reference triangles $(R_k)_{k=1,6}$ (see figure 4). We obtain :

$$a(i,j) = \sum_{k=1}^{6} \int_{R_k} \nabla\psi_i^{(k)}.\nabla\psi_j^{(k)} det J^{(k)} \partial X \partial Y \tag{17}$$

where $J^{(k)}$ is the Jacobian of the affine transformation which applies T_k to R_k and $\psi_j^{(k)}$ are the form functions associated to the reference triangles.

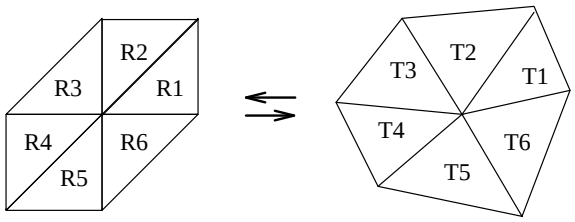

Figure 4: *Support of the basis functions.*

These computations are made at the beginning of the process. Next we solve iteratively the linear system (10). At each iteration the external forces vector L is updated. This system is solved with a conjugate gradient method [FR64]. This technique is well adapted for sparse matrices since it requires only a procedure which applies a vector to the matrix. Secondly this method is iterative and takes also advantage of the initialization.

4 Experimental Results

4.1 A synthetic example

Figure 5 shows a first example obtained on a synthetic image. We consider a very simple image which represents a black rectangle into a white background. The second image is obtained by application of a global affine transformation to the first image. Notice that the mean displacement issued from this transformation is about 3.5 pixel. Next we initialized the adaptive mesh by a regular

Distance Errors (Pixel)	*All Nodes* (225 Nodes)	*High Gradient Nodes* (42 Nodes)	*High Curvature Nodes* (4 Nodes)
Min Value	0.0462	0.1100	0.1100
Max Value	1.2100	0.7930	0.3500
Mean Value	0.5330	0.4110	0.2250
Standard Deviation	0.3120	0.1720	0.0854

Table 1: Statistics on the distance between the real corresponding point and the corresponding point issued from our algorithm. These statistics are presented first for all nodes and next for the most significant nodes.

19×19 mesh. After the mesh adaption stage, we initialized the displacement field in the border of the mesh with the real transformation. We computed the displacement field and next we measured some quantitative parameters which illustrate the accuracy of the results (see table 1). For each node (except on the border where the result is correct by construction), we computed the distance between its real corresponding point and the corresponding point obtained with our algorithm. We presented some statistics for all the nodes, for high gradient nodes and for high curvature nodes.

This example allows us to compare results obtained with a regular mesh and results obtained with an adaptive mesh. Results are very similar particularly near contour points but the computational time is approximatively reduced by a factor sixteen, since we have taken one node each four pixels.

4.2 An articulated structure: the knee

We experimented our method on medical images. We processed first a sequence of images showing the motion of a knee. The patient is lying down and only his leg moves. Images are sagittal views provided with a Magnetic Resonance Imagery device. The resolution of the images is 256×256. The quality of these images is very good (see Figure 7) but some contours are too thin.

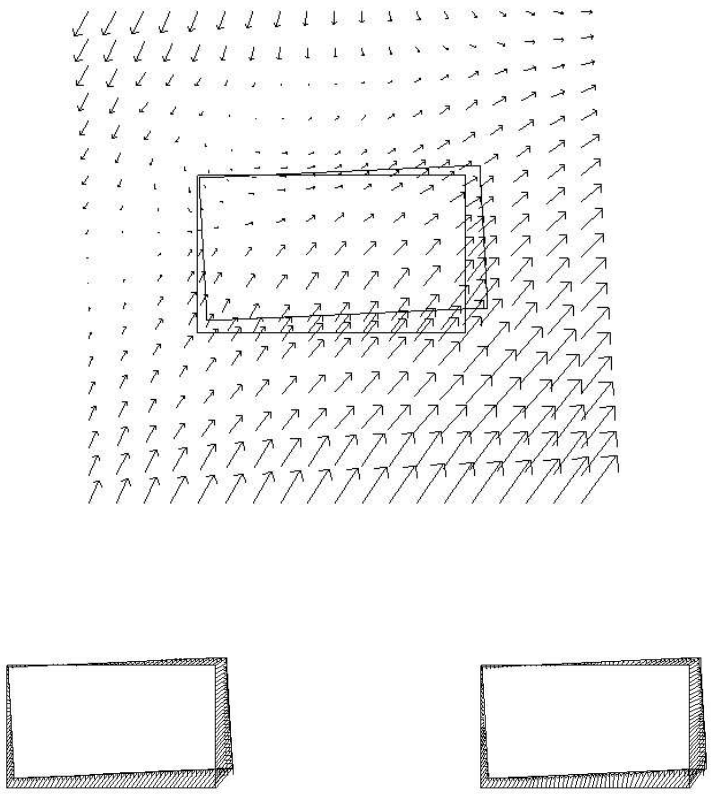

Figure 5: *Synthetic example : Rectangle image and global affine transformation. Up, displacement field computed with the adaptive mesh. Bottom left, we present the contour points of the two images and the non rigid motion field associated to contour points computed by interpolation from the deformed mesh. Bottom right, we present the same result obtained with a regular mesh. The accuracy of the displacement field is preserved.*

Figure 6: *Gradient norm of the grey level function for a frame of the knee sequence. The contours on the front part of the leg below the knee are too thin and so the gradient norm tends to vanish.*

It is a problem for our algorithm which needs the computation of the image gradient. We used Canny-Deriche filter to compute derivatives. The contour model for the Canny-Deriche filter is the step edge model and so it fails for such thin contours. Figure 6 shows the gradient norm of the image function for an image of the sequence. One can see that the front part of the leg below the knee is not well detected. A second problem concerns the presence of artifacts on bottom and upper parts of the images. The last important point is the time resolution. The displacement between two successive frames is relatively high. We have measured approximatively that the displacement length of the contour points exceeds sometimes ten pixels.

Figure 7 shows two successive frames of the knee sequence. We applied our deformation mesh technique on the first frame. We initialized the mesh by considering one node each five pixels. We present the result of the adaptation of the mesh. The result is fine except on the front part of the leg below the knee. We have next solved the motion equation associated to this mesh and we present the final displacement vector field. It allows to appreciate the role played by the regularization term. The smoothness constraint allows some points to move as they are not salient. It is not true unfortunately for the

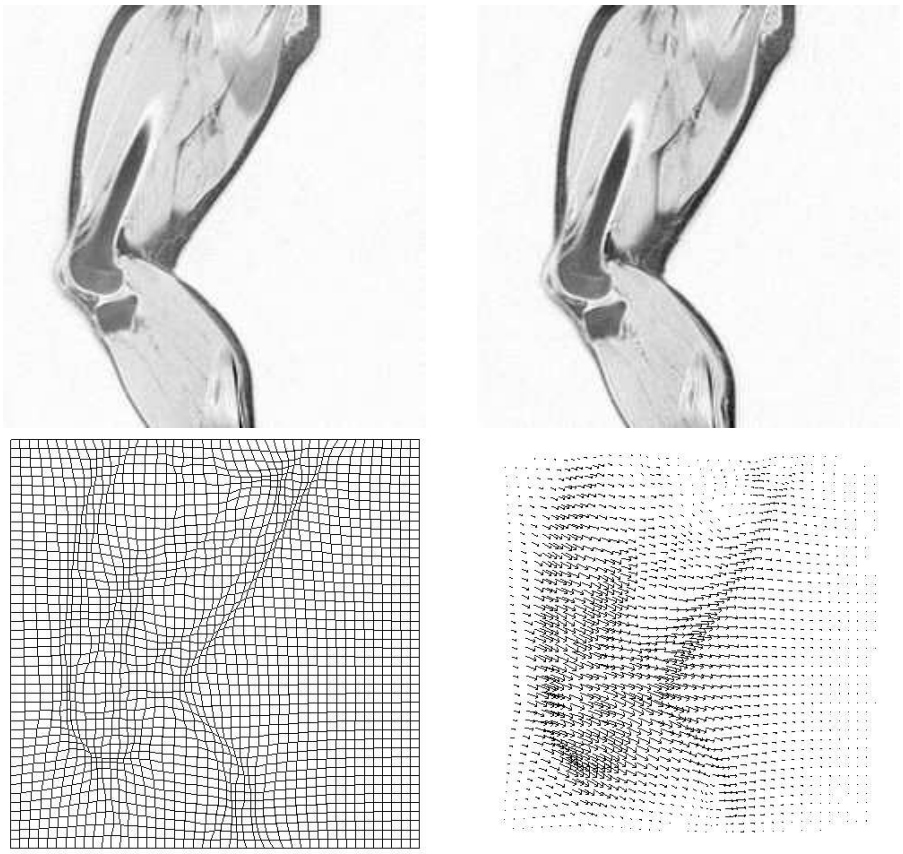

Figure 7: *Up, two successive frames of the knee sequence. Bottom left, adaptive mesh computed on the first image. Bottom right, displacement field between the two images.*

lower part of the leg. In this case this is all the region which has no salient point. In the future we will have to consider stronger smoothness constraints to improve the results.

We present next results which allow to appreciate qualitatively the accuracy of the motion field. Once we have the displacement vector field over the whole mesh we compute by interpolation the motion field over the whole image. Thus we can reconstruct a frame with the next frame on the sequence and the motion field computed between the two frames. The algorithm of reconstruction is straightforward :

- Consider a pixel p on the first frame.
- Apply the corresponding displacement vector to obtain a real point on the second image.
- Compute its grey level by interpolation and affect this value to p.

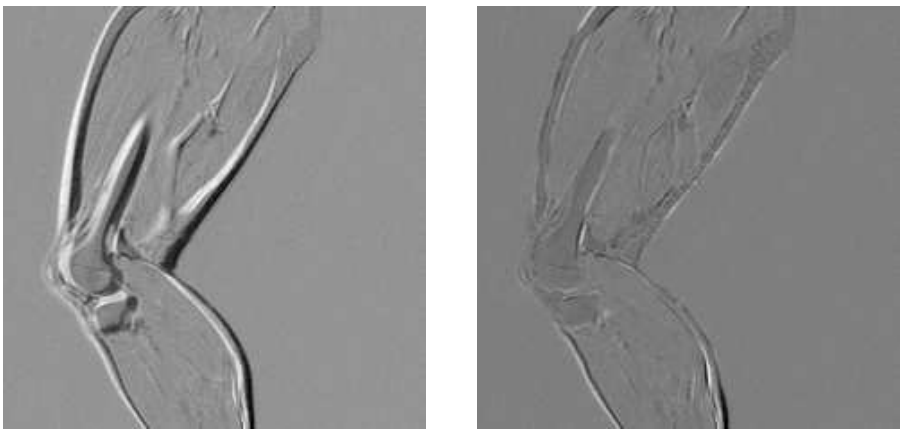

Figure 8: *Left, subtraction of two successive frames of the knee sequence. Right, subtraction of the first frame with a reconstruction of the same image computed with the displacement field.*

Once we have done this we can subtract the real image with the reconstructed image. Figure 8 shows the difference between the two successive frames and

next the difference between the first frame and the same frame after reconstruction. The background grey level of the images corresponds to a null difference. Since the luminance varies during the motion, subtracting the image and the reconstructed image shows some imperfections. On the other hand overlaying the reconstructed image with the real contour points matches perfectly as it is shown in Figure 9. Such a result was expected since our method is based on high gradient norm points matching.

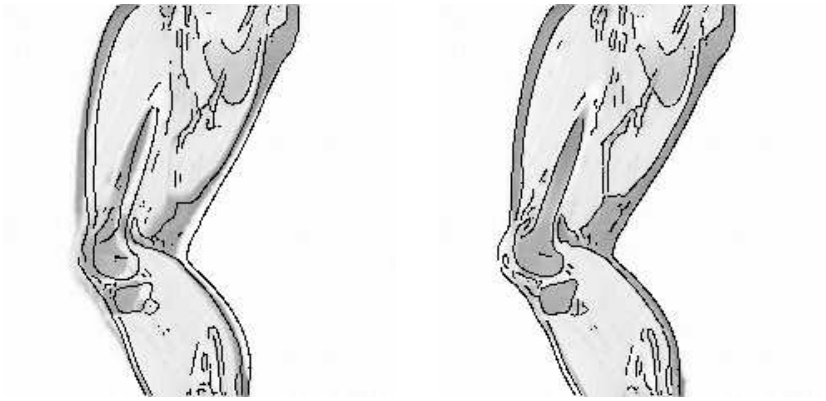

Figure 9: *Left, superposition of a reconstructed image from the previous image in the sequence with the real contour points of the previous image. Right, superposition of the same reconstructed image with the corresponding real contour points. The superposition is almost perfect.*

The reconstruction algorithm can use any sequence of motion fields. Thus we have considered three successive frames of the knee sequence. We have reconstructed the first frame with the third frame and the two motion fields. Figure 10 shows super-imposition of the reconstructed image with real contour points.

4.3 A deformable structure: the heart

Next we have experimented our method on a sequence of heart angiographic images. The resolution of the images is also 256×256. These images are much

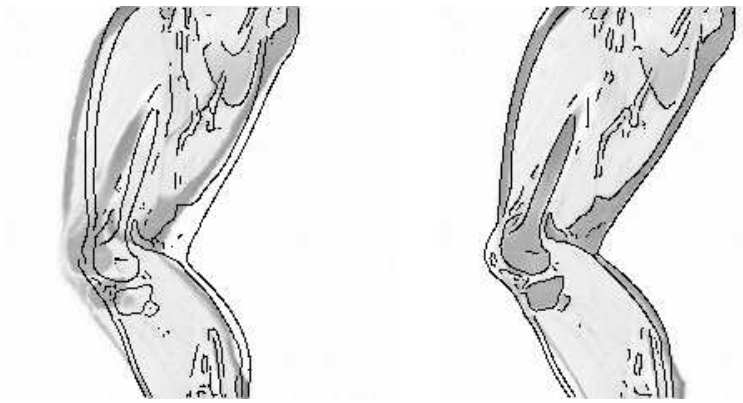

Figure 10: *Left, superposition of a reconstructed image from the antepenultimate image in the sequence with the real contour points of this image. Right, superposition of the same reconstructed image with the corresponding real contour points. The reconstruction used also two successive motion fields. The result remains excellent except on the extreme bottom part of the image.*

better because the boundaries of the heart are everywhere compatible with the step edge model. The time resolution is also better and the displacement lengths between successive images do not exceed five or six pixels.

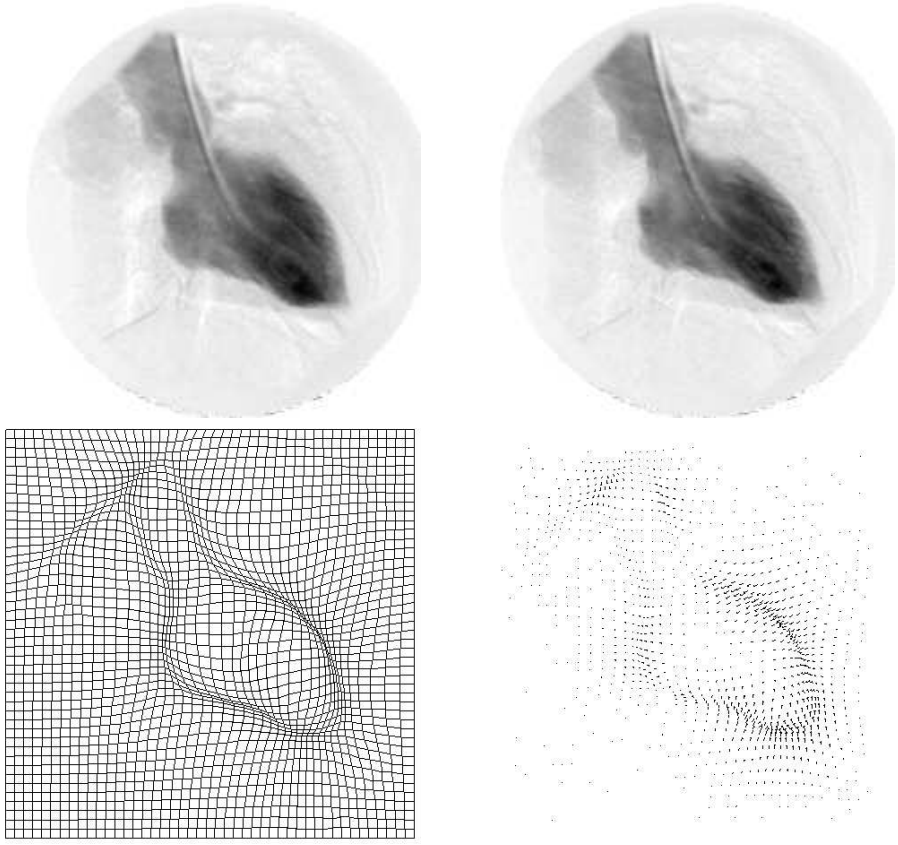

Figure 11: *Up, two successive frames of the heart sequence. Bottom left, adaptive mesh computed on the first image. Bottom right, displacement field between the two images.*

Figure 11 shows two successive frames, the result of the adaptation of the mesh for the first frame and the corresponding motion field. We see so clearly

that the heart contracts himself. We applied the reconstruction algorithm for a sequence of six successive images. Figure 12 shows results of super-imposition.

4.4 $3D$ case:preliminary results

Figure 13 shows preliminary results of the extension of our method for the $3D$ case. We implemented the mesh deformation technique. We used a 20-neighborhood to enforce the stability of the mesh. We experimented first this process for a volumetric image of a dark cube into a clear background. The figure 13 presents an isosurface of the cube and the associated deformed cube. The deformation is a global affine transformation. The adaption stage is initialized by a regular $15 \times 15 \times 15$. Next we computed the displacement field. The computation of the motion is based on a straightforward extension of the method presented for the $2D$ case. The main difference is that we used the total curvature K_T instead of the Euclidean curvature. Recall that for a surface point, K_T is the mean of the two principal curvatures [MBF92].

We used the real transformation to initialize the displacement field in the border of the mesh. The accuracy of results is proved by the numerical results presented on table 2. Notice that the mean displacement issued from the affine transformation is about 3.5 voxel.

We used next our algorithm for a real medical image. We processed volumetric data of a human heart. The dimensions of these images are $48 \times 64 \times 48$. We present the isosurface corresponding to the boundaries of the heart and the corresponding adaptive mesh. For the clarity of the presentation we show the mesh just near high gradient norm points (Figure 14).

Conclusion and Future Research

We described a method to compute displacement fields in a sequence of temporal images. It consists in minimizing an energy which takes into account differential similarities for contour points and imposes a regularity constraint everywhere. In order to reduce numerical complexity, we introduced a structured adaptive mesh. This approach allows to speed-up the algorithm and preserves the accuracy of the results. We have presented results on $2D$ synthe-

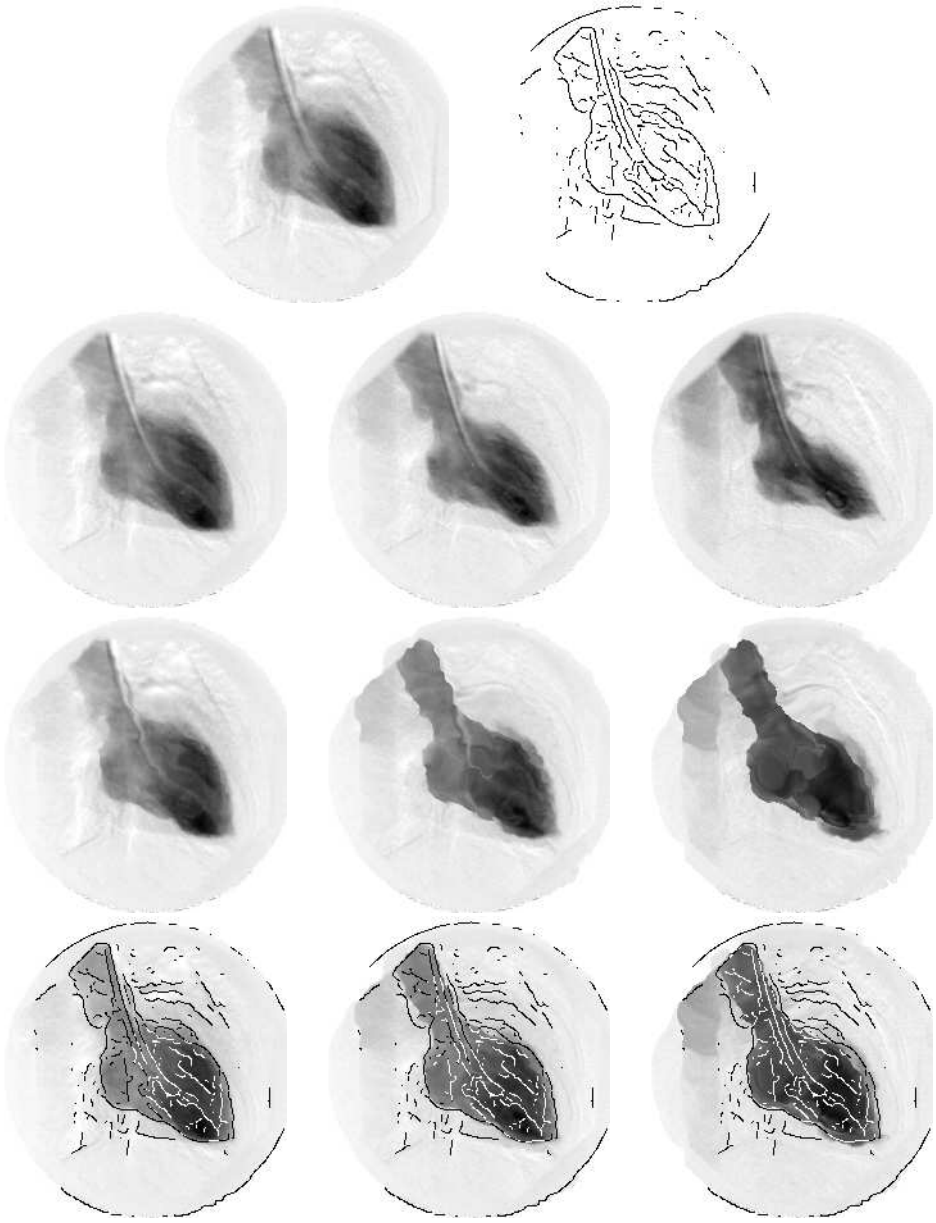

Figure 12: **First line:** *Frame two of the heart sequence and corresponding contour points image.* **Second line:** *Frames three, six and nine of the sequence.* **Third line:** *Reconstruction of the frame two from respectively the frames three, six and nine.* **Fourth line:** *These same images are displayed with the contour points of the frame two.*

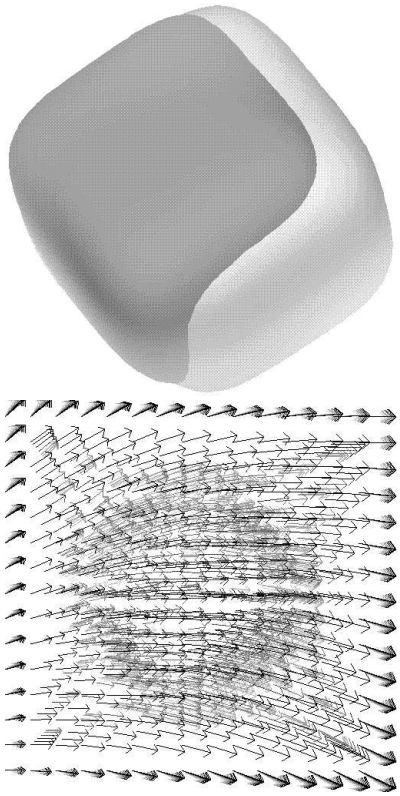

Figure 13: *Up isosurface of a cube image (dark grey) with the corresponding deformed cube (bright grey). Bottom, displacement field corresponding to the deformation.*

Distance Errors (Voxel)	*All Nodes* (1331 Nodes)	*High Gradient Nodes* (427 Nodes)	*High Curvature Nodes* (147 Nodes)
Min Value	0.0321	0.0321	0.0619
Max Value	1.4400	1.4400	0.8510
Mean Value	0.4720	0.4190	0.4380
Standard Deviation	0.2380	0.1780	0.1540

Table 2: Statistics on the distance between the real corresponding points and the corresponding points issued from our algorithm.

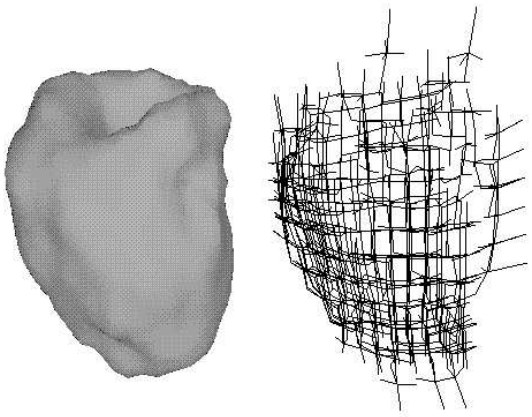

Figure 14: *Left, isosurface of the boundaries of a heart. Right, the deformed mesh is presenting just near high gradient norm points.*

tic and real images and a preliminary $3D$ example. Future work will pursue the study of temporal sequences of $3D$ images.

Acknowledgments

We thank Hervé Guillard and Eric Bardinet for many fruitful discussions. We thank also Jean-Philippe Thirion who suggested us the reconstruction algorithm . We thank finally Dr. Boisgard of the G. Montpied Hospital in Clermont Ferrand who provided the knee images.

This work was partially supported by **Digital Equipment Corp**. The knee images were acquired in collaboration with Siemens.

References

[AD92] A.A. Amini and J.S. Duncan. Bending and stretching models for left ventricle wall motion analysis from curves and surfaces. *Image and Vision Computing*, 10(6):418–430, July/August 1992.

[And86] D.A. Anderson. Constructing adaptive grids with poisson grid generators. In *Conference on Numerical Grid Generation in Computational Fluid Dynamics*, pages 125–136, LandShut, West Germany, July 1986.

[Aya93] N. Ayache. *Volume Image Processing, Results and Research Challenges.* September 1993. INRIA Research Report (2050).

[BAC94a] S. Benayoun, N. Ayache, and I. Cohen. Adaptive meshes and non rigid motion computation. In *International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994.

[BAC94b] S. Benayoun, N. Ayache, and I. Cohen. Mouvement de structures $2d$ déformables : utilisation de caractéristiques géométriques différentielles. In *Congrès de Reconnaissance des Formes et Intelligence Artificielle*, pages 589–600, Paris, France, Janvier 1994.

[BK89] Ruzena Bajcsy and Stane Kovacic. Multiresolution elastic matching. *Computer Vision Graphics and Image Processing*, 46:1–21, 1989.

[CAS92] Isaac Cohen, Nicholas Ayache, and Patrick Sulger. Tracking points on deformable objects using curvature information. In *Proceedings of the Second European Conference on Computer Vision 1992*, pages 458–466, Santa Margherita Ligure, Italy, May 1992. In Lecture Notes in Computer Science: Computer Vision – ECCV92, Vol. 588 Springer-Verlag.

[CHA94] C.W. Chen, T.S. Huang, and M. Arrott. Modeling, analysis and visualization of left ventricle shape and motion by hierarchical decomposition. *IEEE Transactions on PAMI*, 16(4):342–356, April 1994.

[Cia87] P.G. Ciarlet. *The finite element methods for elliptic probems.* North-Holland, Amsterdam, 1987.

[DG93] J.A. Desideri and H. Guillard. *Méthodes numériques pour la combustion supersonique, Bilan et perspectives.* Aout 1993. INRIA Research Report (2002).

[DWS83] L.S. Davis, Z. Wu, and H. Sun. Contour-based motion estimation. *Computer Vision, Graphics and Image Processing*, 23:313–326, 1983.

[FR64] R. Fletcher and C.M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, pages 149–154, 1964.

[HG93a] W.-C. Huang and D.B. Goldgof. Adaptive-size meshes for rigid and nonrigid shape analysis and synthesis. *IEEE Transactions on PAMI*, 15(6):611–616, June 1993.

[HG93b] W.C. Huang and D.B. Goldgof. Non-rigid motion analysis using nonlinear finite element modeling. In *S.P.I.E.*, volume 2031, 1993. Geometric Methods in Computer Vision II.

[KG94] S. Kumar and D. Goldgof. Automatic tracking of spamm grid and the estimation of deformation parameters from cardiac mr images. *IEEE Transactions on Medical Imaging*, 13(1):122–132, March 1994.

[MBF92] O. Monga, S. Benayoun, and O.D. Faugeras. Using partial derivatives of 3d images to extract typical surface features. In *Proceedings CVPR '92, Urbana Champaign, Illinois.* IEEE, July 1992. also an INRIA Research Report (1599).

[PD83] B. Palmerio and A. Dervieux. *2D and 3D unstructured mesh adaption relying on physical analogy.* September 1983. Universit de Nice Research Report (207).

[SD92] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on PAMI*, 14(11):1061–1075, November 1992.

[TH94] L.A. Tang and T.S. Huang. Quantifying facial expressions : smiles. In *Proceedings of the Workshop Journee INRIA Analyse/Synthèse d'Images*, pages 22–27, Paris, France, Janvier 1994.

[TV91] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *Proceedings CVPR '91, Lahaina, Maui, Hawai*, pages 70–75. IEEE, June 1991.

[WTK87] A. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *International Journal of Computer Vision*, pages 133–144, 1987.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399